

REMARKS

Claims 1-24 are pending in this application.

Specification

Per item 4 in the office action, the Abstract was objected to because it states, "One embodiment disclosed relates to" Applicants have hereby amended the Abstract in accordance with the Examiner's suggestion to remove this language. Hence, Applicants respectfully submit that this objection is now overcome.

Per item 5 in the office action, the disclosure was objected to because of the informalities of sentence fragments on page 5. Applicants have hereby amended the first paragraph on page 5 to correct the informalities in a manner which is compatible with the flow chart of FIG. 2A. No new matter is being added. Hence, Applicants respectfully submit that this objection is now overcome.

Per item 2 in the office action, the Abstract was also objected to because it allegedly does not include the technical disclosure of the improvement over the previous method of process deactivation. Applicants respectfully traverse this objection.

Applicants respectfully submit that the Abstract recites a combination of technical elements which comprise an improvement over the previous method of process deactivation. For example, the Abstract recites, "The process-wide deactivation operation is called by outstanding threads of the process when the outstanding threads re-enter the operating system's kernel." Applicants respectfully submit that the aforementioned technical element is an improvement over the previous method of process deactivation.

35 USC 112

Claims 10-12 were rejected under 35 USC 112, first paragraph, as allegedly containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Applicants respectfully traverse this rejection.

Regarding claim 10, claim 10 recites “determining if the process is still deactivatable.” This step corresponds to block 230 in FIG. 2B of the application. As described in the specification, “a further determination **230** is made as to whether the process is still considered **by the swapper** to be deactivatable.” (Page 5, lines 25-27, emphasis added in part.) In addition, the specification further describes as follows. “Process deactivation is a mechanism by which a process’ threads are forcibly stopped and that process’ memory pages are marked to be pushed out from more rapidly accessible semiconductor memory to more slowly accessible disk memory. Processes are deactivated in order to free up resources on a heavily loaded system. For example, processes may be deactivated when the system is thrashing or under severe memory pressure. Deactivating a process frees up resources by: a) allowing reserved memory areas (for example, one user structure or UAREA per thread) of the process to be pushed out to disk by the virtual memory handler; b) stopping that process from running or attempting to run; and c) stopping the process from generating input/output traffic due to page faults. This allows more deserving (non-deactivated) processes to use the freed-up resources to complete their tasks.” (Page 3, lines 14-25.)

As is apparent from the above-cited description, the swapper (which swaps memory between memory chips and disk storage) considers the process to still be deactivatable or to be not deactivatable. Therefore, applicants respectfully submit that one of ordinary skill in the pertinent art is enabled by the description in the specification to be able to query or otherwise determine from the swapper whether the process is still considered to be deactivatable.

Regarding claim 11, claim 11 recites that “if the process is still deactivatable, then finishing the process-wide deactivation operation and turning off a flag for the

process.” These steps correspond to blocks 232 and 234 in FIG. 2B of the application. As described in the specification, “If the process is still deactivatable, then, in accordance with one embodiment, the process-wide deactivation operation is finished **232**, the SLOAD flag is turned off **234** to implement deactivation of the process, and the procedure returns **236** with an indication that deactivation was successful (immediately).” (Page 5, lines 27-31.) Therefore, applicants respectfully submit that one of ordinary skill in the pertinent art is enabled by the description in the specification to be able to finish the process-wide deactivation operation and turn off a flag for the process.

Regarding claim 12, claim 12 recites that “if the process is no longer deactivatable, then unsuspending threads of the process and finishing the process-wide deactivation operation.” These steps correspond to blocks 238 and 240 in FIG. 2B of the application. As described in the specification, “Otherwise, if the process is no longer deactivatable, then all the process’ threads that have been suspended are un-suspended **238**, the process-wide deactivation operation is finished **240**, and the procedure returns **242** with an indication of a failure to deactivate the process. (Page 5, line 31 through page 6, line 2.) Therefore, applicants respectfully submit that one of ordinary skill in the pertinent art is enabled by the description in the specification to be able to unsuspend threads of the process and finish the process-wide deactivation operation.

For at least the above-discussed reasons, applicants respectfully submit that this rejection is now overcome.

35 USC 103

Original claims 1, 23 and 24 were rejected under 35 U.S.C. § 103 as being unpatentable over Kawahara et al (USP 6,910,209, hereinafter “Kawahara”). Applicants respectfully traverse this rejection.

*I. THE **THREAD TERMINATION** TECHNIQUE PER KAWAHARA DOES NOT DISCLOSE OR SUGGEST A METHOD OF **PROCESS DEACTIVATION***

Applicants respectfully submit that the **thread termination** technique per Kawahara does not disclose or suggest the method of claim 1 which pertains to **process deactivation**. There are many substantial differences between thread termination and process deactivation such that a method for the former cannot be simply adapted to become a method for the latter.

A. *THREADS AND PROCESSES ARE TECHNOLOGICALLY DISTINCT SUCH THAT A TECHNIQUE FOR TERMINATING A THREAD IS NOT READILY ADAPTABLE TO DISACTIVATING A PROCESS*

First of all, applicants respectfully submit that **threads and processes are technologically distinct** such that a technique applied to threads is very unlikely to technologically work when applied to processes. For example, in most multithreading operating systems, a process is given its **own memory address space**, while a thread is not given its own memory address space. Hence, threads typically share the memory heap belonging to their parent process. As another example, a process may have **many threads**.

Therefore, for at least this reason, applicants respectfully submit that it was not and is not obvious to one of ordinary skill in the art that terminating a **thread** per Kawahara may be readily adapted to deactivating a **process** per claim 1. If this rejection is maintained, applicants respectfully request an explanation of how terminating a thread per Kawahara can be readily adapted to deactivate a multi-threaded process per claim 1.

B. *KAWAHARA TEACHES A TECHNIQUE FOR FORCIBLY TERMINATING A THREAD WHILE CLAIM 1 RECITES STOPPING (NOT TERMINATING) THREADS AS PART OF A METHOD OF DEACTIVATING (AGAIN, NOT TERMINATING) A PROCESS*

Kawahara teaches a technique for forcibly **terminating** a thread. As discussed in the Abstract of Kawahara, "Apparatus, methods, and computer program products are disclosed for a process of terminating a thread in a clean, certain, and forcible manner. A thread is **forcibly terminated** in such a manner that data structures in the system are not left in an inconsistent state and the overall system status is not damaged." (Emphasis added.)

In contrast, claim 1 recites **stopping (not terminating) threads** as part of a method of **deactivating (again, not terminating) a process**. As described in the specification, deactivating a process **pushes out that process' memory pages from rapidly accessible memory** (typically, semiconductor memory) **to more slowly accessible memory** (typically, disk memory). This is stated in the specification as follows. "Process deactivation is a mechanism by which a process' threads are forcibly stopped and that process' memory pages are marked to be pushed out from more rapidly accessible semiconductor memory to more slowly accessible disk memory. Processes are deactivated in order to free up resources on a heavily loaded system. For example, processes may be deactivated when the system is thrashing or under severe memory pressure. Deactivating a process frees up resources by: a) allowing reserved memory areas (for example, one user structure or UAREA per thread) of the process to be pushed out to disk by the virtual memory handler; b) stopping that process from running or attempting to run; and c) stopping the process from generating input/output traffic due to page faults. This allows more deserving (non-deactivated) processes to use the freed-up resources to complete their tasks." (Page 3, lines 14-25.)

Therefore, for at least this reason, applicants respectfully submit that the **deactivation** of a multi-threaded process per claim 1 is patentably distinct over the **termination** of a thread per Kawahara. If this rejection is maintained, applicants respectfully request an explanation of why the Examiner believes that thread termination reads onto process deactivation.

II. KAWAHARA DOES NOT DISCLOSE OR SUGGEST INITIATING A PROCESS-WIDE DEACTIVATION OPERATION

Applicants respectfully submit that Kawahara does not disclose or suggest "initiating a **process-wide deactivation** operation" (emphasis added) per claim 1.

In relation to this claim element, the office action cites to column 2, lines 43-44 and column 5, lines 16-18. However, these citations to Kawahara relate **to thread termination, not process deactivation**. As discussed above, thread termination and process deactivation are patentably distinct operations. In addition, there is no teaching or suggestion in Kawahara of a deactivation operation which is **process-wide**.

Therefore, for at least these further reasons, applicants respectfully submit that claim 1 is patentably distinguished over Kawahara.

*III. KAWAHARA DOES NOT DISCLOSE OR SUGGEST DETERMINING WHETHER THREADS OF THE PROCESS ARE CURRENTLY **SUSPENDABLE***

Applicants respectfully submit that Kawahara does not disclose or suggest “determining whether threads of the process are currently **suspendable**” (emphasis added) per claim 1.

In relation to this claim element, the office action cites to column 7, lines 11-14 and column 5, lines 16-18. However, these citations to Kawahara relate to thread **termination, not suspension**. Suspension does not kill a thread, while termination does kill a thread.

Therefore, for at least this further reason, applicants respectfully submit that claim 1 is patentably distinguished over Kawahara.

*IV. KAWAHARA DOES NOT DISCLOSE OR SUGGEST MOVING THE THREADS OF THE PROCESS THAT ARE CURRENTLY SUSPENDABLE TO A **STOPPED STATE***

Applicants respectfully submit that Kawahara does not disclose or suggest “moving the threads of the process that are currently suspendable to a **stopped state**” (emphasis added) per claim 1.

In relation to this claim element, the office action cites to column 6, lines 45-47 and column 5, lines 16-18. However, these citations to Kawahara relate to **terminating a thread, not stopping a thread**. Unlike thread termination, putting a thread in a stopped state does not kill the thread.

Therefore, for at least this further reason, applicants respectfully submit that claim 1 is patentably distinguished over Kawahara.

*V. KAWAHARA DOES NOT DISCLOSE OR SUGGEST THAT THE **PROCESS-WIDE DEACTIVATION OPERATION IS CALLED BY OUTSTANDING THREADS***

***OF THE PROCESS WHEN THE OUTSTANDING THREADS RE-ENTER A
KERNEL OF THE OPERATING SYSTEM***

Applicants respectfully submit that Kawahara does not disclose or suggest that “the **process-wide deactivation operation is called** by outstanding threads of the process **when the outstanding threads re-enter a kernel** of the operating system.” (Emphasis added.) This claim element is expressly recited in claim 1.

In relation to this claim element, the office action cites to column 5, lines 8-9 and column 5, lines 16-18. However, these citations to Kawahara relate to **individual thread termination, not a process-wide deactivation operation.**

Furthermore, there appears to be no disclosure of **outstanding threads re-entering a kernel in the citations**. There also appears to be no disclosure of calling a process-wide deactivation operation when the outstanding threads re-enter the kernel.

Therefore, for at least these further reasons, applicants respectfully submit that claim 1 is patentably distinguished over Kawahara.

Claims 23 and 24 are independent claims that recite some of the same (or similar) limitations as discussed above in relation to claim 1. Hence, applicants respectfully submit that claims 23 and 24 overcome this rejection for some of the same (or similar) reasons as discussed above in relation to claim 1.

Original claims 2-22 were rejected under 35 U.S.C. § 103 as being unpatentable over Kawahara in view of one or more other references. Applicants respectfully traverse these rejections.

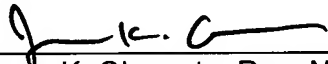
Claims 2-22 depend from claim 1. Hence, applicants respectfully submit that claims 2-22 are patentably distinguished over combinations based on Kawahara for at least the reasons discussed above in relation to claim 1.

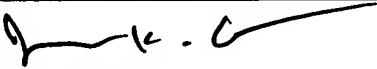
Conclusion

For the above-discussed reasons, applicant respectfully submits that the objections and rejections of the latest office action are now overcome. Favorable action is respectfully requested.

Respectfully Submitted,

Dated: March 3, 2008


James K. Okamoto, Reg. No. 40,110
Tel: (408) 436-2111
Fax: (408) 436-2114

CERTIFICATE OF MAILING			
I hereby certify that this correspondence, including the enclosures identified herein, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.			
Signature:			
Typed or Printed Name:	James K. Okamoto	Dated:	March 3, 2008
Express Mail Mailing Number (optional):			